

Chương 11

MA TRẬN CĂN BẢN

1. VÍ DỤ DẪN NHẬP 1

- Bài toán: Viết chương trình thực hiện các yêu cầu sau bằng phương pháp lập trình hướng đối tượng
 - + Nhập ma trận một chiều các số nguyên
 - + Xuất ma trận một chiều các số nguyên
 - + Tính tổng các giá trị trong ma trận
- Chương trình

1. VÍ DỤ DẪN NHẬP 1

```
11. #include <iostream.h>
12. class CMaTranNguyen
13. {
14.     private:
15.         int a[100][100];
16.         int m, n;
17.     public:
18.         void Nhap( );
19.         void Xuat( );
20.         int TinhTong( );
21. } ;
```

1. VÍ DỤ DẪN NHẬP 1

```
11. void main( )
12. {
13.     CMaTranNguyen a;
14.     a.Nhap( ) ;
15.     a.Xuat( ) ;
16.     int kq = a.TinhTong( ) ;
17.     cout<<"Tong = " <<kq;
18. }
```

1. VÍ DỤ DẪN NHẬP 1

```
11. void CMaTranNguyen::Nhap( )
12. {
13.     cout<<"Nhap m : ";
14.     cin>>m;
15.     cout<<"Nhap n : ";
16.     cin>>n;
17.     for (int i=0 ; i<m ; i++)
18.         for (int j=0; j<n; j++)
19.         {
20.             cout<<"a["<<i<<
                "]"<<j<<" ]:";
21.             cin>>a[i][j];
22.         }
23. }
```

1. VÍ DỤ DẪN NHẬP 1

```
11. void CMatranNguyen::Xuat ( )
12. {
13.     for (int i=0 ; i<m ; i++)
14.     {
15.         for(int j=0; j<n; j++)
16.             cout<<a[i][j]<<" ";
17.         cout<<"\n" ;
18.     }
19. }
```

1. VÍ DỤ DẪN NHẬP 1

```
11. int CMaTranNguyen::TinhTong( )
12. {
13.     int S = 0;
14.     for (int i=0 ; i<m ; i++)
15.         for (int j=0; j<n; j++)
16.             S = S + a[i][j];
17.     return S;
18. }
```

2. VÍ DỤ DẪN NHẬP 2

- Bài toán: Viết chương trình thực hiện các yêu cầu sau bằng phương pháp lập trình hướng đối tượng
 - + Nhập ma trận một chiều các số thực
 - + Xuất ma trận một chiều các số thực
 - + Tìm phần tử lớn nhất của ma trận
- Chương trình

2. VÍ DỤ DẪN NHẬP 2

```
11. #include <stdio.h>
12. #include <conio.h>
13. class CMaTranThuc
14. {
15.     private:
16.         float a[100][100];
17.         int m, n;
18.     public:
19.         void Nhap();
20.         void Xuat();
21.         float LonNhat();
22. } ;
```

2. VÍ DỤ DẪN NHẬP 2

```
11. void main( )
12. {
13.     CMaTranThuc a;
14.     a.Nhap( ) ;
15.     a.Xuat( ) ;
16.     float kq = a.LonNhat( ) ;
17.     printf( "...%f" , kq ) ;
18. }
```

2. VÍ DỤ DẪN NHẬP 2

```
11. void CMaTranThuc::Nhap( )
12. {
13.     printf("Nhap m: ") ;
14.     scanf("%d", &m) ;
15.     printf("Nhap n : ") ;
16.     scanf("%d", &n) ;
17.     for (int i=0 ; i<m ; i++)
18.         for (int j=0; j<n; j++)
19.         {
20.             printf("a[%d][%d]: "
21.                 , i, j) ;
22.             scanf("%f", &a[i][j]) ;
23.         }
```

2. VÍ DỤ DẪN NHẬP 2

```
11. void CMatranThuc::Xuat ( )
12. {
13.     for (int i=0 ; i<m ; i++)
14.     {
15.         for (int j=0; j<n; j++)
16.             printf( "%8.3f" ,a[i][j] ) ;
17.         printf( "\\n" ) ;
18.     }
19. }
```

2. VÍ DỤ DẪN NHẬP 2

```
11. float CMaTranThuc::LonNhat()  
12. {  
13.     float ln = a[0][0];  
14.     for (int i=0 ; i<m ; i++)  
15.         for (int j=0; j<n; j++)  
16.             if (a[i][j] > ln)  
17.                 ln = a[i][j];  
18.     return ln;  
19. }
```

3. VÍ DỤ DẪN NHẬP 3

- Bài toán: Viết chương trình thực hiện các yêu cầu sau bằng phương pháp lập trình hướng đối tượng
 - + Nhập ma trận một chiều các phân số
 - + Xuất ma trận một chiều các phân số
 - + Đếm số lượng giá trị dương có trong ma trận
- Chương trình

3. VÍ DỤ DẪN NHẬP 3

```
11. #include <stdio.h>
12. #include <conio.h>
13. class CPhanSo
14. {
15.     private:
16.         int tu;
17.         int mau;
18.     public:
19.         void Nhap();
20.         void Xuat();
21.         int KTDuong();
22. } ;
```

3. VÍ DỤ DẪN NHẬP 3

```
11. class CMaTranPhanSo
12. {
13.     private:
14.         CPhanSo a[100][100];
15.         int m, n;
16.     public:
17.         void Nhap( );
18.         void Xuat( );
19.         int DemDuong( );
20. } ;
```


3. VÍ DỤ DẪN NHẬP 3

```
11. void main( )
12. {
13.     CMaTranPhanSo a;
14.     a.Nhap( ) ;
15.     a.Xuat( ) ;
16.     int kq = a.DemDuong( ) ;
17.     printf( "...%d: ", kq ) ;
18. }
```

3. VÍ DỤ DẪN NHẬP 3

```
11. void CPhanSo::Nhap( )
12. {
13.     printf("Nhap tu :");
14.     scanf("%d", &tu);
15.     printf("Nhap mau :");
16.     scanf("%d", &mau);
17. }
```

3. VÍ DỤ DẪN NHẬP 3

```
11. void CPhanSo::Xuat ( )  
12. {  
13.     printf( "%d/%d ", tu, mau);  
14. }
```

3. VÍ DỤ DẪN NHẬP 3

```
11. int CPhanSo::KTDuong( )  
12. {  
13.     if (tu*mau > 0)  
14.         return 1;  
15.     return 0;  
16. }
```

3. VÍ DỤ DẪN NHẬP 3

```
11. void CMaTranPhanSo::Nhap( )
12. {
13.     printf("Nhap m : ");
14.     scanf("%d", &m);
15.     printf("Nhap n : ");
16.     scanf("%d", &n);
17.     for (int i=0; i<m; i++)
18.         for (int j=0; j<n; j++)
19.             {
20.                 printf("a[%d][%d]:", i, j);
21.                 a[i][j].Nhap();
22.             }
23. }
```

3. VÍ DỤ DẪN NHẬP 3

```
11. void CMaTranPhanSo::Xuat ( )
12. {
13.     for (int i=0; i<m ; i++)
14.     {
15.         for (int j=0; j<n; j++)
16.         {
17.             a[i][j].Xuat ( ) ;
18.             printf("  ");
19.         }
20.         printf(" \n" ) ;
21.     }
22. }
```

3. VÍ DỤ DẪN NHẬP 3

```
11. int CMaTranPhanSo::DemDuong( )
12. {
13.     int dem = 0;
14.     for (int i=0; i<m ; i++)
15.         for (int j=0; j<n; j++)
16.             if(a[i][j].KTDuong( )==1)
17.                 dem = dem + 1;
18.     return dem;
19. }
```

4. XÂY DỰNG LỚP MA TRẬN SỐ NGUYÊN

- Hãy xây dựng lớp số nguyên với các phương thức như sau:
 - + Phương thức nhập ma trận
 - + Phương thức xuất ma trận
 - + Phương thức liệt kê các giá trị chẵn trong ma trận
 - + Phương thức tính tổng các phần tử trong ma trận
 - + Phương thức tính tổng các giá trị cực đại trong ma trận
 - + Phương thức đếm số lượng giá trị lẻ có trong ma trận
 - + Phương thức đếm số lần xuất hiện của giá trị x trong ma trận

4. XÂY DỰNG LỚP MA TRẬN SỐ NGUYÊN

- Hãy xây dựng lớp số nguyên với các phương thức như sau:
 - + Phương thức kiểm tra ma trận có tồn tại giá trị 0 hay không?
 - + Phương thức kiểm tra ma trận có toàn chẵn hay không?
 - + Phương thức sắp xếp các phần tử trong ma trận tăng dần từ trên xuống dưới và từ trái sang phải

4. XÂY DỰNG LỚP MA TRẬN SỐ NGUYÊN

```
11. class CMaTranNguyen
12. {
13.     private:
14.         int a[100][100];
15.         int m, n;
16.     public:
17.         void Nhap();
18.         void Xuat();
19.         void LietKeChan();
20.         int TinhTong();
21.         int TongCucDai();
22.         int DemLe();
23.         int DemXuatHien(int);
24.         int KTTonTaiKhong();
25.         int KTToanChan();
26.         void SapTang();
27. } ;
```

4. XÂY DỰNG LỚP MA TRẬN SỐ NGUYÊN

```
11. void CMaTranNguyen::Nhap( )
12. {
13.     printf("Nhap m : ");
14.     scanf("%d", &m);
15.     printf("Nhap n : ");
16.     scanf("%d", &n);
17.     for (int i=0; i<m; i++)
18.         for (int j=0; j<n; j++)
19.             {
20.                 printf("a[%d]:", i);
21.                 scanf("%d", &a[i][j]);
22.             }
23. }
```

4. XÂY DỰNG LỚP MA TRẬN SỐ NGUYÊN

```
11. void CMaTranNguyen::Xuat ( )
12. {
13.     for (int i=0; i<m; i++)
14.     {
15.         for (int j=0; j<n; j++)
16.             printf( "%4d", a[i][j] );
17.         printf( "\n" );
18.     }
19. }
```

4. XÂY DỰNG LỚP MA TRẬN SỐ NGUYÊN

```
11. void CMaTranNguyen::LietKeChan()  
12. {  
13.     for (int i=0; i<m; i++)  
14.         for (int j=0; j<n; j++)  
15.             if (a[i][j]%2 == 0)  
16.                 printf("%4d",a[i][j]);  
17. }
```

4. XÂY DỰNG LỚP MA TRẬN SỐ NGUYÊN

```
11. int CMaTranNguyen::TinhTong( )
12. {
13.     int S=0;
14.     for (int i=0; i<m; i++)
15.         for (int j=0; j<n; j++)
16.             S = S+a[i][j];
17.     return S;
18. }
```

4. XÂY DỰNG LỚP MA TRẬN SỐ NGUYÊN

```
11. int CMaTranNguyen::LonNhat ( )
12. {
13.     int ln=a[0][0];
14.     for (int i=0; i<m; i++)
15.         for (int j=0; j<n; j++)
16.             if (a[i][j] > ln)
17.                 ln = a[i][j];
18.     return ln;
19. }
```

4. XÂY DỰNG LỚP MA TRẬN SỐ NGUYÊN

```
11. int CMaTranNguyen::TongCucDai ( )
12. {
13.     int S=0;
14.     int ln=LonNhat ( );
15.     for (int i=0; i<m; i++)
16.         for (int j=0; j<n; j++)
17.             if (a[i][j]==ln)
18.                 S = S+a[i][j];
19.     return S;
20. }
```


4. XÂY DỰNG LỚP MA TRẬN SỐ NGUYÊN

```
11. int CMaTranNguyen::DemLe ( )
12. {
13.     int dem=0;
14.     for (int i=0; i<m; i++)
15.         for (int j=0; j<n; j++)
16.             if (a[i][j]%2 == 1)
17.                 dem = dem+1;
18.     return dem;
19. }
```

4. XÂY DỰNG LỚP MA TRẬN SỐ NGUYÊN

```
11. int CMaTranNguyen::  
    DemXuatHien(int x)  
12. {  
13.     int dem=0;  
14.     for (int i=0; i<m; i++)  
15.         for (int j=0; j<n; j++)  
16.             if (a[i][j] == x)  
17.                 dem = dem+1;  
18.     return dem;  
19. }
```

4. XÂY DỰNG LỚP MA TRẬN SỐ NGUYÊN

```
11. int CMaTranNguyen::  
    KTTonTaiKhong( )  
12. {  
13.     for (int i=0; i<m; i++)  
14.         for (int j=0; j<n; j++)  
15.             if (a[i][j] == 0)  
16.                 return 1;  
17.     return 0;  
18. }
```

4. XÂY DỰNG LỚP MA TRẬN SỐ NGUYÊN

```
11. int CMaTranNguyen::KTToanChan( )
12. {
13.     for (int i=0; i<m; i++)
14.         for (int j=0; j<n; j++)
15.             if (a[i][j]%2 == 1)
16.                 return 0;
17.     return 1;
18. }
```

4. XÂY DỰNG LỚP MA TRẬN SỐ NGUYÊN

```
11. int CMaTranNguyen::SapTang( )
12. {
13.     for (int k=0; k<m*n-2; k++)
14.         for (int l=k+1; l<n*n-1; l++)
15.             if (a[k/n][k%n]>a[l/n][l%n])
16.                 {
17.                     int temp = a[k/n][k%n];
18.                     a[k/n][k%n]=a[l/n][l%n];
19.                     a[l/n][l%n] = temp;
20.                 }
21. }
```

5. XÂY DỰNG LỚP MA TRẬN SỐ THỰC

- Hãy xây dựng lớp số thực với các phương thức như sau:
 - + Phương thức nhập ma trận
 - + Phương thức xuất ma trận
 - + Phương thức tính tổng các phần tử trong ma trận
 - + Phương thức tính tổng các giá trị cực tiểu trong ma trận
 - + Phương thức đếm số lần xuất hiện của giá trị x trong ma trận

5. XÂY DỰNG LỚP MA TRẬN SỐ THỰC

- Hãy xây dựng lớp số thực với các phương thức như sau:
 - + Phương thức kiểm tra ma trận có tồn tại giá trị 0 hay không?
 - + Phương thức sắp xếp ma trận tăng dần từ trên xuống dưới và từ trái sang phải.

5. XÂY DỰNG LỚP MA TRẬN SỐ THỰC

```
11. class CMaTranThuc
12. {
13.     private:
14.         float a[100][100];
15.         int m, n;
16.     public:
17.         void Nhap();
18.         void Xuat();
19.         float TinhTong();
20.         float NhoNhat();
21.         float TongCucTieu();
22.         int DemXuatHien(float);
23.         int KTTonTaiKhong();
24.         void SapTang();
25. } ;
```


5. XÂY DỰNG LỚP MA TRẬN SỐ THỰC

```
11. void CMaTranThuc::Nhap( )
12. {
13.     printf("Nhap n : ");
14.     scanf("%d", &n);
15.     for (int i=0; i<m; i++)
16.         for (int j=0; j<n; j++)
17.             {
18.                 printf("a[%d]:", i);
19.                 scanf("%f", &a[i][j]);
20.             }
21. }
```

5. XÂY DỰNG LỚP MA TRẬN SỐ THỰC

```
11. void CMaTranThuc::Xuat( )
12. {
13.     for (int i=0; i<m; i++)
14.     {
15.         for (int j=0; j<n; j++)
16.             printf( "%8.3f", a[i][j] );
17.         printf( "\n" );
18.     }
19. }
```

5. XÂY DỰNG LỚP MA TRẬN SỐ THỰC

```
11. float CMaTranThuc::TinhTong( )
12. {
13.     float S=0;
14.     for (int i=0; i<m; i++)
15.         for (int j=0; j<n; j++)
16.             S = S+a[i][j];
17.     return S;
18. }
```

5. XÂY DỰNG LỚP MA TRẬN SỐ THỰC

```
11. float CMaTranThuc::NhoNhat()  
12. {  
13.     float nn=a[0][0];  
14.     for (int i=1; i<m; i++)  
15.         for (int j=0; j<n; j++)  
16.             if (a[i][j] < nn)  
17.                 nn = a[i][j];  
18.     return nn;  
19. }
```

5. XÂY DỰNG LỚP MA TRẬN SỐ THỰC

```
11. float CMaTranThuc::TongCucTieu()  
12. {  
13.     float S=0;  
14.     float nn=NhoNhat();  
15.     for (int i=0; i<m; i++)  
16.         for (int j=0; j<n; j++)  
17.             if (a[i][j]==nn)  
18.                 S = S+a[i][j];  
19.     return S;  
20. }
```

5. XÂY DỰNG LỚP MA TRẬN SỐ THỰC

```
11. int CMaTranThuc::  
    DemXuatHien(float x)  
12. {  
13.     int dem=0;  
14.     for (int i=0; i<m; i++)  
15.         for (int j=0; j<n; j++)  
16.             if (a[i][j] == x)  
17.                 dem = dem+1;  
18.     return dem;  
19. }
```

5. XÂY DỰNG LỚP MA TRẬN SỐ THỰC

```
11. int CMaTranThuc::  
    KTTonTaiKhong( )  
12. {  
13.     for (int i=0; i<m; i++)  
14.         for (int j=0; j<n; j++)  
15.             if (a[i][j] == 0)  
16.                 return 1;  
17.     return 0;  
18. }
```

4. XÂY DỰNG LỚP MA TRẬN SỐ THỰC

```
11. void CMaTranThuc::SapTang( )
12. {
13.     for (int k=0; k<m*n-2; k++)
14.         for (int l=k+1; l<m*n-1; l++)
15.             if (a[k/n][k%n]>a[l/n][l%n])
16.                 {
17.                     float temp=a[k/n][k%n];
18.                     a[k/n][k%n]=a[l/n][l%n];
19.                     a[l/n][l%n]= temp;
20.                 }
21. }
```